Ontology-based Modular Architecture for Surgical Autonomous Robots

R. Perrone[1], F. Nessi[1], E. De Momi[1], F. Boriero[2], M. Capiluppi[2], P. Fiorini[2], G. Ferrigno[1]

[1]*Politecnico di Milano, Department of Electronics, Information and Bioengineering (DEIB), Milano, Italy*
-{roberta.perrone, elena.demomi, giancarlo.ferrigno}@polimi.it, federico.nessi@mail.polimi.it
[2]*Dipartimento di Informatica, Università di Verona, Strada le Grazie, Verona, Italy*
-{fabrizio.boriero, marta.capiluppi, paolo.fiorini}@univr.it

INTRODUCTION

Medical robotics is becoming a leading application area in which the surgical assistance provided by robots is rapidly rising such as the Intuitive® Telesurgery System for the Da Vinci Robot, [1] and the NOTES robot for a semi-autonomous surgical task execution, [2]. However, the complexity of the medical environment (which is a dynamic environment) has been a major barrier, preventing a wider use of robotic technology. Providing robots with cognitive architecture for specialized tasks requires a systematic approach in describing the knowledge needed for reaching a goal. At the same time, knowledge processing allows writing of more general and flexible control programs. In this work, we present a workflow for the design and the deployment of an architecture for the execution of a surgical task (i.e. tool positioning on the correct trajectory for needle insertion), where the architecture's components skeleton are automatically derived from ontological description. We formalized basic knowledge in a way that is readable and processable by both man and machine. Ontologies can describe abstract things, like workflows or tasks (e.g. manipulation or grasping [3]) but also real things such as devices [4]. Moreover, the robot control system is able to autonomously perform the initialization of the considered surgical procedure. The system manages also possible failures of software and hardware components.

MATERIALS AND METHODS

We used a 7 degrees of freedom robot (KUKA LWR4+, Germany) in a target approaching task [5], for example the preliminary phase of a surgical needle insertion. The robot movement is followed by two trackers (NDI Optotrack® Certus, NDI Polaris Vicra®) with different characteristics (i.e. accuracy, sampling rate, field of view). Since the operating room is quite crowed during surgery, the field of view of a tracker may be occluded. We considered a second tracker as backup, so that if the best tracker is no more able to provide the position of the robot, we can rely on a second tracker. For the sake of the safety, we set for every tracker two levels of velocity (i.e. *fast* and *slow*), so that when the robot reaches a *Critical Area* (i.e. the sphere around the head of the patient), the robot slows down. The setup is shown in **Fig. 1**. The design of the architecture is made using knowledge stored in an ontology, written in OWL[1], that provides the description of components (e.g. ports, types of data exchanged and priority of sensors). The ontology is built exploiting already existing upper ontology (SUMO[2]) that allows us to append new classes and instances to more general classes like "Device" and "Agent".

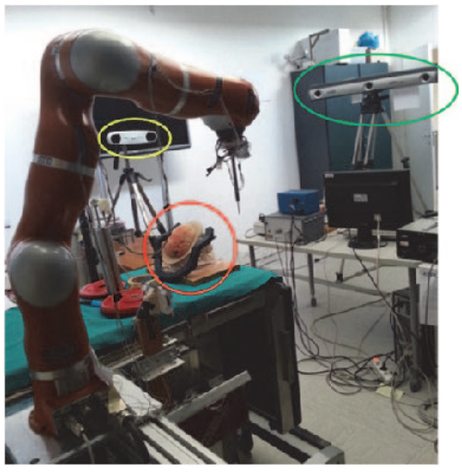

Fig. 1. Scenario of the case study. The end-effector of the 7 dof robot is seen by two optical trackers that are highlighted in the yellow and green circles, respectively. The red circle represents the intra-operative head frame.

The ontology is populated with all devices used at our laboratory. As in the KNOWROB[3] project, we used Prolog to query the ontology and retrieve the knowledge about components (e.g. type of ports, data exchanged, priority), then we processed the results to obtain XML mapping. The XML is the basis for the development of the robotic architecture, achieved using OROCOS[4] running in a ROS environment. A high-level control is implemented, with a Sensor Manager (SM) that reads the data coming from trackers and chooses at each moment the best available tracker according to the performance of the single sensor to track both the robot and the intra-operative reference frame and the sensor accuracy in providing tracking data. This information is

[1] http://www.w3.org/TR/owl-ref/
[2] http://www.ontologyportal.org/
[3] http://www.knowrob.org/
[4] http://www.orocos.org/

sent via ORoCoS connection to a Supervisor component, that tells the robot the target to be reached and constantly reads the robot position provided by the tracking system, comparing to the provided target. Besides the high-level control, a simple middle-level control (e.g. a position interpolator) is provided in order to control the robot movement. The deployment of the architecture is accomplished with a Coordinator-Configurator pattern (following what is presented in [6]) using a pure Lua module[5]. The Coordinator component is a hierarchical finite-state machine (see **Fig. 2**) derived from the Task Ontology and written using rFSM Tool[6] with the knowledge about the task to be performed. Events raised from the Supervisor component assert the transition from one state to another, requiring a specific components topology to the Configurator. This allows changing the entire system architecture in reaction to a single event.

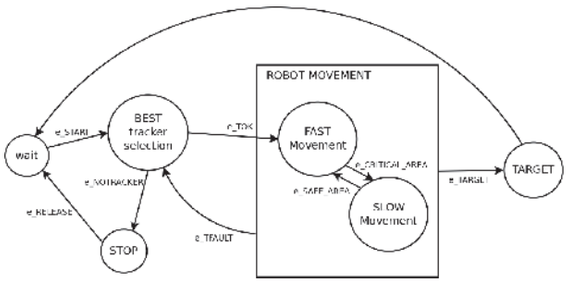

Fig. 2. Coordinator Finite State Machine. It represents the workflow used to drive the robot during target approach.

In order to evaluate the architecture performance, we simulated different emergency situations and measured the latencies of the architecture response through ROS Time Stamps. First, we simulated the fault of a single tracker (*Sensor Swap* test) by occluding the field of view (FOV), in such a way that it is no more able to provide the position of the robot. The latency from the fault recognition and the connection of the best available tracker was measured. Then, we simulated the contemporary fault of all the trackers (*No Sensor To Stop* test) by occluding their FOV. The latency between the recognition of the unavailability of the sensors and the STOP command sent to the robot was measured. Each test was performed 50 times. Mean and standard deviation were computed from the acquired data and used to evaluate the architecture's performance.

RESULTS

The results of performed tests are represented in **Fig. 3**. The *No Sensor To Stop* test resulted to perform better (a mean of 6.6 *ms*), due to less computational time. Considering the worst case of the robot moving at its maximum speed (20 *cm/s*) this leads to a mean of 1.3 *mm* of untracked movement before the STOP assertion. The *Sensor Swap* test had worse performance (a mean of 12 *ms*) because of the need to create a connection with the new best available tracker.

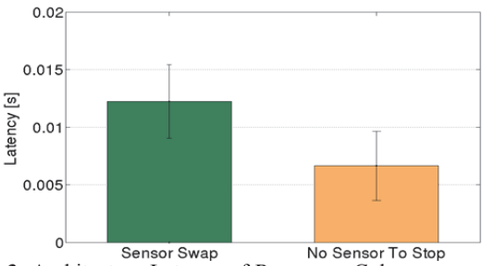

Fig. 3. Architecture Latency of Response. Columns represent mean and standard deviation of the measured latencies (50 repetitions each)

DISCUSSION

In this work we presented an ontology-based approach for the development of an autonomous surgical robotic architecture. The advantages are the standardization of the knowledge on the hardware components and the modularity (in fact it is possible to append every kind of sensor with minimal change on the code). The high-level control provides the possibility to manage possible faults that may happen during the execution. Future works may be related to the fusion of different kinds of knowledge: for example, the information of a Kinect® will be useful to handle security problems related to the presence of too many subjects in the surgical FOV.

ACKNOWLEDGEMENTS

This work is supported within the EuRoSurge European project (FP7-ICT-2011-7-288233).

REFERENCES

[1] Guthart G.S., Salisbury J.K., The Intuitive Telesurgery System: Overview and Application. Proceedings of 2000 IEEE, International Conference on Robotics and Automation. 2000.

[2] Dumpert J., Lehman A.C. et al. Semi-autonomous surgical task using a miniature In vivo Surgical Robot, 31st Annual International Conference of the IEEE EMBS, 2009.

[3] Varadarajan K. M., and Vincze M., Ontological knowledge management framework for grasping and manipulation. IROS Workshop: Knowledge Representation for Autonomous Robots. 2011.

[4] Mudunuri R., Burgert O., and Neumuth T. "Ontological Modelling of Surgical Knowledge." GI Jahrestagung. 2009. pp. 1044-1054.

[5] De Momi E., Perrone R., Capiluppi M., et al., EuRoSurge Workflow: From ontology to surgical task execution, 3rd Workshop on New Technologies for Computer Assisted Surgery. 2013.

[6] Klotzbucher M., Biggs G., Bruyninckx H., "Pure Coordination using the Coordinator–Configurator Pattern". 3rd International Workshop on Domain-Specific Languages and models for ROBotic systems.

[5] https://bitbucket.org/kmarkus/dng
[6] http://people.mech.kuleuven.be/~mklotzbucher/rfsm